Machine-Learning-powered Algorithmic Trading

Bachelor Degree in Informatics Engineering
School of Informatics of Barcelona – UPC BarcelonaTECH

Hermes Valenciano Farré
hermes.valenciano@est.fib.upc.edu

Director: Lluís A. Belanche Muñoz

January 24, 2019

Abstract

The problem addressed by this work is asset price prediction and automation of a trading system. The focus is on comparing traditional trading strategies with the predictions made by a Recurrent Neural Network. The motivation for this project came from the questions: Can Deep Learning models be used to forecast the prices of a traded asset so that it is viable for trading? And if so, how large would the competitive advantage be with respect to other simpler trading strategies? Would it be sufficient to beat the market on a regular basis? These and other questions meet their corresponding answer in this work. The methodology used for collecting the results has been to compare the results that three trading strategies obtain with respect to a Machine Learning approach when trading the currency pair EUR/USD in the period contained between 2000 and 2018, using in each case the most favorable set of parameters. These results have revealed that one-step-ahead prediction forecasts based on historical price data alone still remain below standards and therefore are not particularly useful for trading.

Contents

List of Figures

List of Tables

1 Context

1.1 Introduction

In the era of immediate information, financial markets move unusually rapidly and tendencies spread all over the globe in a matter of minutes. The amount of data that is available for analize (past prices, news feeds, fundamental data) is too big for a human to use it efficiently in its decision-making. But with the continous development of faster machines and algorithms, a major part of these tasks can be done automatically and much more effectively.

The fact that computers can help humans with data-related problems is well known and widely used. Nonetheless, not until we entered the 21st century, did the market volume of algorithmic trading went from a modest 15% to almost 90% in less than fifteen years [7], as shown in Figure 1. The big actors of the markets (institutional investors, funds, banks, etc.), unlike retail investors (people that manage and invest its own assets), rely on huge distributed systems to carry out its activities (risk management, portfolio optimization, investment strategies development, etc.). In this research project, the role of Machine Learning models in the field of asset price prediction will be discussed, and the results of the implementation of a trading strategy using ML will be compared to the ones obtained with traditional rule based strategies.

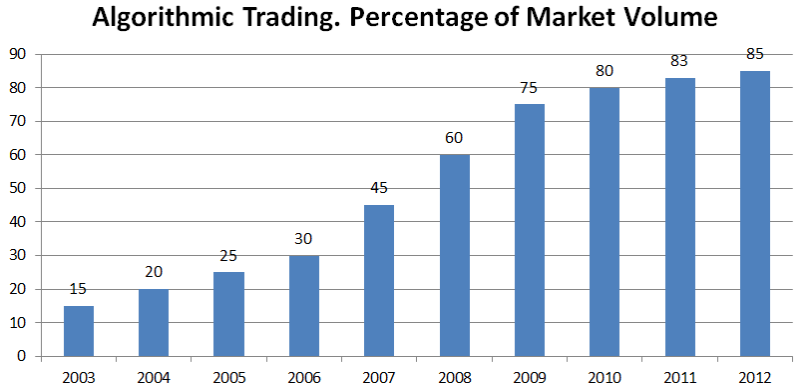

Figure 1: *Algorithmic Trading. Percentage of Market Volume.[7]*

1.2 Concepts

1.2.1 Machine Learning

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to progressively improve their performance on a specific task. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.[17]

1.2.2 Trading Strategy

Trading strategies are methods that traders use to determine when to buy and sell assets in the financial markets. Strategies may be based on technical analysis, fundamental analysis, quantitative methods, or a combination of decision factors.

1.2.3 Automated Trading System

Automated trading systems (ATS), also referred to as mechanical trading systems, algorithmic trading, automated trading or system trading, allow traders to establish specific rules for both trade entries and exits that, once programmed, can be automatically executed via a computer. The trade entry and exit rules can be based on simple conditions such as a moving average crossover, or they can be complicated strategies that require a comprehensive understanding of the programming language specific to the user's trading platform, or the expertise of a qualified programmer. Here are presented some advantadges of ATS[19] when compared to classic manual placing orders:

- **Minimizes Emotions**. Automated trading systems minimize emotions throughout the trading process. By keeping emotions in check, traders typically have an easier time sticking to the plan. Since trade orders are executed automatically once the trade rules have been met, traders will not be able to hesitate or question the trade. In addition to helping traders who are afraid to "pull the trigger," automated trading can curb those who are apt to overtrade – buying and selling

at every perceived opportunity.

- **Ability to Backtest**. Backtesting applies trading rules to historical market data to determine the viability of the idea. When designing a system for automated trading, all rules need to be absolute, with no room for interpretation (the computer cannot make guesses – it has to be told exactly what to do). Traders can take these precise sets of rules and test them on historical data before risking money in live trading. Careful backtesting allows traders to evaluate and fine-tune a trading idea, and to determine the system's expectancy – i.e., the average amount that a trader can expect to win (or lose) per unit of risk.

- **Preserves Discipline**. Because the trade rules are established and trade execution is performed automatically, discipline is preserved even in volatile markets. Discipline is often lost due to emotional factors such as fear of taking a loss, or the desire to eke out a little more profit from a trade. Automated trading helps ensure that discipline is maintained because the trading plan will be followed exactly. In addition, "pilot error" is minimized; for instance, an order to buy 100 shares will not be incorrectly entered as an order to sell 1,000 shares.

- **Achieves Consistency**. One of the biggest challenges in trading is to plan the trade and trade the plan. Even if a trading plan has the potential to be profitable, traders who ignore the rules are altering any expectancy the system would have had. There is no such thing as a trading plan that wins 100% of the time – losses are a part of the game. But losses can be psychologically traumatizing, so a trader who has two or three losing trades in a row might decide to skip the next trade. If this next trade would have been a winner, the trader has already destroyed any expectancy the system had. Automated trading systems allow traders to achieve consistency by trading the plan.

- **Improved Order Entry Speed**. Since computers respond immediately to changing market conditions, automated systems are able to generate orders as soon as trade criteria are met. Getting in or out of a trade a few seconds earlier can make a big difference in the trade's outcome. As soon as a position is entered, all other orders are automatically generated, including protective stop losses and profit targets. Markets can move quickly, and it is demoralizing to have a trade reach the profit target or blow past a stop-loss level – before the orders can even be entered. An automated trading system prevents this from happening.

Automated trading systems boast many advantages, but there are some downfalls and realities[19] that traders should be aware of:

- **Mechanical failures**. The theory behind automated trading makes it seem simple: Set up the software, program the rules and watch it trade. In reality, however, automated trading is a sophisticated method of trading, yet not infallible. Depending on the trading platform, a trade order could reside on a computer – and not a server. What that means is that if an internet connection is lost, an order might not be sent to the market. There could also be a discrepancy between the "theoretical trades" generated by the strategy and the order entry platform component that turns them into real trades.

- **Monitoring**. Although it would be great to turn on the computer and leave for the day, automated trading systems do require monitoring. This is due do the potential for technology failures, such as connectivity issues, power losses or computer crashes, and to system quirks. It is possible for an automated trading system to experience anomalies that could result in errant orders, missing orders, or duplicate orders. If the system is monitored, these events can be identified and resolved quickly.

- **Over-optimization**. Though not specific to automated trading systems, traders who employ backtesting techniques can create systems that look great on paper and perform terribly in a live market. Over-optimization refers to excessive curve-fitting that produces a trading

plan that is unreliable in live trading. It is possible, for example, to tweak a strategy to achieve exceptional results on the historical data on which it was tested. Traders sometimes incorrectly assume that a trading plan should have close to 100% profitable trades or should never experience a drawdown to be a viable plan. As such, parameters can be adjusted to create a "near perfect" plan – that completely fails as soon as it is applied to a live market.

1.2.4 Backtesting System

Backtesting[20] assesses the viability of a trading strategy by discovering how it would play out with historical data.
Backtesting allows a trader to simulate a trading strategy using historical data to generate results and analyze risk and profitability before risking any actual capital.

A well-conducted backtest that yields positive results assures traders that the strategy is fundamentally sound and is likely to yield profits when implemented in reality. A well-conducted backtest that yields suboptimal results will prompt traders to alter or reject the strategy.

Particularly complicated trading strategies, such as strategies implemented by automated trading systems, rely heavily on backtesting to prove their worth, as they are too arcane to evaluate otherwise.

The ideal backtest chooses sample data from a relevant time period of a duration that reflects a variety of market conditions. In this way, one can better judge whether the results of the backtest represent a fluke or sound trading.

1.2.5 Survivorship Bias

One can suffer the effects of survivorship bias when drawing conclusions from an incomplete set of data because that data has ‘survived’ some selection criteria.

In finance, survivorship bias is the tendency for failed companies to be excluded from performance studies because they no longer exist. It often causes the results of studies to skew higher because only companies which were successful enough to survive until the end of the period are

included. For example, a mutual fund company's selection of funds today will include only those that are successful now. Many losing funds are closed and merged into other funds to hide poor performance.[21]

1.3 Stakeholders

In the next sections, the planned stakeholders of this project will be explained.

1.3.1 Project developer

The project developer is the person responsible of defining the goals and scope for the project, as well as developing it. The project developer for this project is Hermes Valenciano, a final-year Computer Science student.

1.3.2 Project director

The project director is the person responsible of guiding the project developer and offering advice both in organization and in the technical details. The director of this project is Lluís Belanche, professor and researcher at UPC.

1.3.3 Audience

Anyone interested in quantitative finance and/or applications of ML can serve as audience for this project.

1.3.4 Users

Researchers trying to build market models could use the work conducted here as a baseline.

1.3.5 Beneficiaries

In particular, those retail investors that use a well defined strategy but execute it periodically by hand, may see a powerful benefit in automating the most of the process. Also, any institution that offers portfolio management

(Goldman Sachs, JP Morgan, HSBC, Santander, etc.) could include this kind of trading in its services, at almost no additional cost.

2 Problem formulation

The problem that concerns this project is the development of an Automated Trading System that uses Machine Learning for its decision-making. For this purpose, there are various key points. The crucial ones are defined here:

2.1 Market selection

Although the stock market is probably the first we think of when it comes to investment, stocks are not the only asset one can trade. There are other assets, traded on other markets, normally with lower transaction fees, that can serve the purpose of being the field of action of the trading agent. Here are shown some examples:

2.1.1 Foreign Exchange

The forex market is a financial market where currencies are traded. This financial market is the most liquid market in the world, as cash is the most liquid of assets. The interbank market is the financial system that trades currency between banks.[4]

2.1.2 Stock Exchange

Stock markets allow investors to buy and sell shares in publicly traded companies. They are one of the most vital areas of a market economy as they provide companies with access to capital and investors with a slice of ownership in the company and the potential of gains based on the company's future performance.[4]

2.1.3 Commodities Market

A commodity market is a physical or virtual marketplace for buying, selling and trading raw or primary products, and there are currently about

50 major commodity markets worldwide that facilitate investment trade in approximately 100 primary commodities.

Commodities are split into two types: hard and soft commodities. Hard commodities are typically natural resources that must be mined or extracted (such as gold, rubber and oil), whereas soft commodities are agricultural products or livestock (such as corn, wheat, coffee, sugar, soybeans and pork).[5]

2.2 Data gathering

In order to train the models that may be wanted to carry out the guiding of the trading strategy, the access to financial data will need to be reliable. It is vital to correctly select the sources of information that will feed the system. Thus, a pro/cons analysis of the alternatives will be required.

2.3 Data preprocessing

This step is optional and will include all those transformations that may be needed to perform to the data before feeding any model. For instance, there are models that are more sensible to the scale, and there are models that appreciate that the provided data is already normalized and/or standardized.

2.4 Definition of the trading strategy

This is the most important step. It is here where it will take place the most extensive part of the experimentation. This step consists of defining the set of rules that the system will use to make its predictions and act consequently.

3 Goals of the project

Here are listed the defined goals for this project:

- Explore the existing possibilities in the intersection between Quantitative Trading and Machine Learning

- Develop a trading strategy that uses Machine Learning for its decision-making
- Compare the existing rule-based trading strategies with the data-driven trading strategies that will be developed
- Build an Automated Trading System that implement the methods developed and can operate autonomously

4 State-of-the-art

4.1 Trading strategies

In the field of investing strategies, it is specially hard to know where the knowledge limits are. This is because any advance done by quantitative finance analysts that is really significative can easily be sold to big corporations and institutions for their private usage only. Thus, any element susceptible of meaning a competitive advantadge will be protected and remain propietary.

4.2 Automated Trading Systems

Early form of Automated Trading System has been used by financial managers and brokers, software based on algorithms. These kind of software were used to automatically manage clients' portfolios. But first service to free market without any supervision from financial advisers and managers to serve clients directly was given in 2008 with the launch of Betterment by Jon Stein. Since then this system is getting improved with development in IT industry, now Automated Trading Systems are managing huge assets all around the globe. As of 2014, more than 75 percent of the stock shares traded on United States exchanges (including the New York Stock Exchange and NASDAQ) originate from automated trading system orders. ATSs can be based on a predefined set of rules which determine when to enter an order, when to exit a position and how much money to invest in each trading product. Trading strategies differ; some are designed to pick market tops and bottoms, others to follow a trend, and others involve complex strategies including randomizing orders to make them less visible in the marketplace. ATSs allow a trader to execute

orders much quicker and manage their portfolio easily by automatically generating protective precautions.[6]

5 Scope of the project

5.1 Scope

This project consists of creating an automated trading system based on Machine Learning. The project will be completed by February 2019. It will include all the required infrastructure to perform the automated trading itself, and the experimentation conducted in order to prove the potential benefits that ML has to offer.

5.2 Methodology and rigor

5.2.1 First part: Research

During the first month (October) of the project, the focus will be on obtaining the largest amount possible of information about already existing trading strategies and the manner in which they could be improved with ML. To effectively complete the following two parts, it is essential to research the way in which ML can be applied in the best form. In order to conduct this research, a number of books and online resources will be consulted and compared.

5.2.2 Second part: Implementation

During the next month (November) all the infrastructure for the automated trading system will be built. This concerns all the data obtention, pipelining and automated processes. Furthermore, the use of an existing Backtesting platform instead of building one specifically for this project will be evaluated and discussed. The methodology that will be used in this part of the project will be iterative. The first thing to do will be to get a prototype that provides basic functionality and then add more features until reaching the desired complexity.

5.2.3 Third part: Experimentation

Once the platform is built, the experimentation will take place occupying the two months left (December and January). In this phase, the main objective is to try out different models and evaluate them against a benchmark. The methodology employed to do the experiments will be the usual one: For each experiment, the hypothesis will be stated, and the used data will be specified. Also, the procedure followed to perform the experiment will be clearly described. Finally, the obtained results will be discussed and (if applicable) compare them with the expected ones.

5.3 Verification and validation

In order to validate the results obtained during the experimentation phase, two mechanisms will be used. The first one is to have regular meetings with the director of the project and let him evaluate the correctness of the conducted work. The other suggested method to ensure that the obtained results are significant is Back-testing of the trading strategy. Back-testing consists on running the trading strategy against past data and see how it performs. This is not a metric to measure future performance, since we cannot assume that what happened on the past will repeat on the future. But for the scope of this project is good enough.

5.4 Possible obstacles and risks

5.4.1 Biased data

One clear obstacle we may have to overcome is the bias of the data. When it comes to financial data, it is usual to have a particular kind of bias known as 'survivorship bias'. This, in stocks data for instance, means that only the companies that are still alive have their stocks' past prices on the majority of databases. But those which bankrupted are normally removed entirely, so there is some valuable information in them that it is not easy to find.

5.4.2 Insufficient data

Another limitation that the project may have is that the models can suffer from data starvation, if there is not enough data to feed them. This obstacle

can normally be overcame buying data packs that come with plenty of data but are not accessible for free.

5.4.3 Bugs

During the implementation part, bugs may occur. But for solving this issue, tests will be made for every module of the system.

5.4.4 Scheduling issues

Finally, we may end up having to face that some part of the project has taken more time than the scheduled. In this case, the proposed solution is to work more hours than the initially scheduled, or in a limit case, to re-schedule the parts.

6 Schedule

6.1 Estimated project duration

The estimated duration for the development of this project is 4 months, starting on October 2018, and finishing on January 2019, as shown in Figure 2.

6.2 Considerations

As it is a research project, it may be possible that the initial plan changes and we have to reconsider some of the technologies that we are using, as well as, the algorithms.

From 24th of December until 2nd of January, the project will stop for a week during Christmas holidays.

7 Project planning

The problem that concerns this project is the development of an Automated Trading System that uses Machine Learning for its decision-making. The different tasks to achieve in order to complete the project are here defined, as well as the resources needed and the precedence constraints.

7.1 Market selection

7.1.1 Task description

Although the stock market is probably the first we think of when it comes to investment, stocks are not the only asset one can trade. There are other assets, traded on other markets, normally with lower transaction fees, that can serve the purpose of being the field of action of the trading agent.

7.1.2 Time dedication

The amount of hours that is planned to require this task is 40: 20 for the Developer and 20 for the Project Director.

7.1.3 Identified subtasks

1. Learn about different markets and its pros/cons for Algorithmic trading [14 hours each]
2. Make a comparative study between the available alternatives [4 hours each]
3. Decide finally in which market will take place the trading activity [1 hour each]

7.1.4 Identified precedence constraints

This task has no identified precedence constraints.

7.1.5 Resources needed

The resources that are going to be needed are:

- Human Resources
  - **Project Director**: The person who has to offer guidance and ensure the technical quality of the project
  - **Developer**: The person who has to do the research and the comparative study
- Hardware Resources

  - **Personal Computer**: MacBook Pro 2016

- Software Resources
  - **macOS High Sierra**: used in all the tasks, expect consulting physical resources as books or encyclopedias
  - **Google Chrome**: used to do the research
  - **LaTeX**: Used to write the documentation

7.2 Data gathering

7.2.1 Task description

In order to train the models that may be wanted to carry out the guiding of the trading strategy, the access to financial data will need to be reliable. It is vital to correctly select the sources of information that will feed the system. Thus, a pro/cons analysis of the alternatives will be required.

7.2.2 Time dedication

The amount of hours that is planned to require this task is 80: 40 for the Developer and 40 for the Project Director.

7.2.3 Identified subtasks

1. Research and compare the available financial data sources and their costs [10 hours each]
2. Decide how often the data will be updated [10 hours each]
3. Build a module that retrieves the data from the internet and stores it in a convenient way [20 hours each]

7.2.4 Identified precedence constraints

This task needs to be done after the market in which the trading will take place is defined, i.e. the task **Market selection** is completed.

7.2.5 Resources needed

- Human Resources
  - **Project Director**: The person who has to offer guidance and ensure the technical quality of the project
  - **Software Developer**: The person who has to compare the available alternatives and build the data gathering module
- Hardware Resources
  - **Personal Computer**: MacBook Pro 2016
- Software Resources
  - **macOS High Sierra**: used in all the tasks
  - **Google Chrome**: used to do the research
  - **Python**: used to build the Data Gathering module
  - **API(s)**: offered by data providers, used to retrieve the data
  - **LaTeX**: used to write the documentation

7.3 Data preprocessing

7.3.1 Task description

This step is optional and will include all those transformations that may be needed to perform to the data before feeding any model. For instance, there are models that are more sensible to the scale, and there are models that appreciate that the provided data is already normalized and/or standardized.

7.3.2 Time dedication

The amount of hours that is planned to require this task is 40: 20 for the Developer and 20 for the Project Director.

7.3.3 Identified subtasks

1. Define the set of transformations that will be applied to the data [5 hours each]

2. Decide whether it is more convenient to store the transformed data or not [5 hours each]

3. Build the Data Preprocessing module using the Python programming language [10 hours each]

7.3.4 Identified precedence constraints

This task must be done after we know what kind of data we will be using. Both the content and the format. Thus, it has to be implemented after the task **Data Gathering**.

7.3.5 Resources needed

- Human Resources
  - **Project Director**: The person who has to offer guidance and ensure the technical quality of the project
  - **Software Developer**: The person who has to implement the Data Preprocessing module

- Hardware Resources
  - **Personal Computer**: MacBook Pro 2016

- Software Resources
  - **macOS High Sierra**: used in all the tasks
  - **Python**: used to build the Data Preprocessing module
  - **LaTeX**: Used to write the documentation

7.4 Definition of the trading strategy

7.4.1 Task description

This is the most important step. It is here where it will take place the most extensive part of the experimentation. This step consists of defining the set of rules that the system will use to make its predictions and act consequently.

7.4.2 Time dedication

The amount of hours that is planned to require this task is 245: 105 for the Project director and 140 for the Developer.

7.4.3 Identified subtasks

1. Select the model (or models) that are going to be used [20 hours]
2. Set-up the backtesting platform to evaluate the performance of the strategy [20 hours]
3. Experiment with the hyperparameters and other configurations [100 hours]

7.4.4 Identified precedence constraints

In order to train the different models and evaluate their performance, the data must be accessible and pre-processed (if applicable). So, this task would need to wait until the **Data Preprocessing** task is completed.

7.4.5 Resources needed

- Human Resources
  - **Project Director**: The person who has to offer guidance and ensure the technical quality of the project
  - **Software Developer**: The person who has to implement the software and make the appropriate decisions
- Hardware Resources
  - **Personal Computer**: MacBook Pro 2016
- Software Resources
  - **macOS High Sierra**: used in all the tasks, expect consulting physical resources, such as books or encyclopedias
  - **Python**: used to build the module
  - **LaTeX**: Used to write the documentation

8 Possible deviations of the schedule

The above sections showed the proposed planning for the project. The available amount of time was divided into the recognized tasks with the assumptions that there will not be any unexpected event.

Nevertheless, we have to identify which unscheduled things can happen, how would they affect the schedule and which is the action plan if they occur.

- **Problem**: The data we decide to use is biased, and due to that the models cannot perform well
  - **Solution**: Buy unbiased data
  - **Effect to the schedule**: 3 more days in the Trading Strategy Development task
- **Problem**: The selected API(s) for obtaining the data are not reliable or take too long to respond
  - **Solution**: Have redundancy in data gathering
  - **Effect to the schedule**: 2 more days in the Data Gathering task
- **Problem**: The models chosen to drive the trading strategy take a huge amount of time to get trained, to the point of making impossible an effective experimentation
  - **Solution**: Split the dataset or buy a cloud computing plan to train the models in the cloud
  - **Effect to the schedule**: 3 more days of development in the Trading Strategy Development task

9 Gantt chart

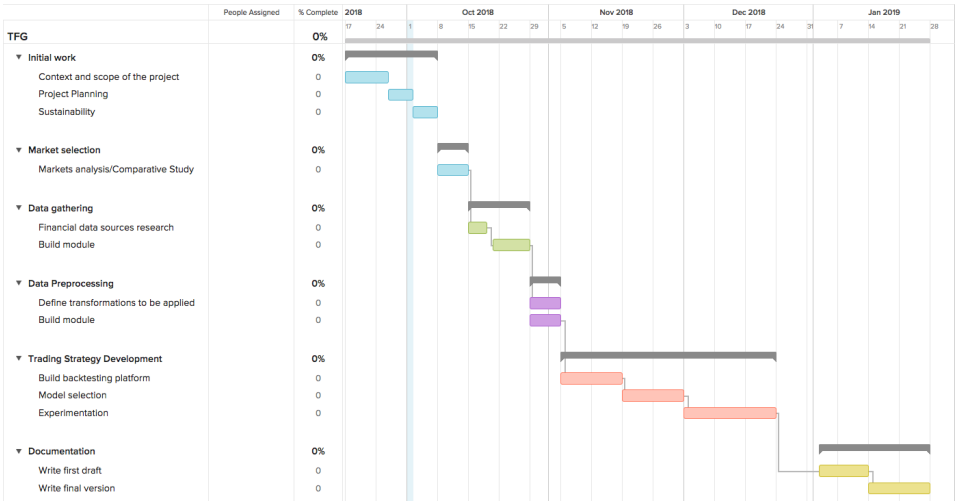

Figure 2: *Gantt chart of the project. Made with TeamGantt[8]*

10 Self-evaluation of sustainability competence

The key insights that I have extracted from the act of answering the survey are mainly two: Firstly, that I am mostly aware of the importance of providing sustainable solutions to real-world problems, and secondly, that currently I'm not capable of certainly tell whether a proposed solution is compliant with the sustainability standards or not.

I think that since Informatics is not an industry that requires a lot of physical resources (generally only computers and electricity), the aspect of providing eco-friendly solutions, for me is not the main concern when analyzing the viability of a project. But it certainly is a fact of major importance if it requires lots of computational power and a big deployment of servers and data centers.

Doing this survey I've realized that the best possible project is not the one which generates more profit, but instead, the one which is capable of providing the world with a respectful product or service that drive us as a species to the common good. Furthermore, it has lightened in me the curiosity to explore the available tools that exist in order to evaluate

the social and environmental impact that the project will have once it is launched and during its life cycle.

To sum up, and in order to discuss my strengths and weaknesses in this field, I'd say that I'm pretty capable to see if a product is empowering the users and democratizing the access to resources or instead is only using them in a profitable and selfish way. On the contrary, I have to admit my ignorance when it comes to seriously break into pieces the problem of sustainability analysis of a project, but it is something that from now on I will consider when starting a project or taking work responsibilities.

11 Cost estimation

11.1 Direct costs

The direct costs of the project will be specified for every one of the main parts of it, as described in the deliverable 2 (Project planning).

The salaries of the different roles associated to every stage of the project have been calculated using a reference website [9].

The amortization for the personal computer has been calculated using this formula:

$$HL * \frac{PL}{YL * 365 * Hday}$$

HL = Hours using the laptop for this project
YL = Useful life in years
Hday = Hours per day using the laptop
PL = Price Laptop

We suppose that the life of our laptops is 6 years and that people usually use the laptop around 8 hours per day.

11.1.1 Market selection

Table 1: *Human resources costs*

Human resources					
Role	**Units**	**Salary/hour (€)**	**Hours/week**	**Weeks**	**Cost (€)**
Project manager	1	35	20	1	700
Market researcher	1	25	20	1	500
TOTAL	**-**	**-**	**-**	**-**	**1200 €**

Table 2: *Hardware resources costs*

Hardware resources				
Item	**Units**	**Price/unit (€)**	**Useful life (years)**	**Amortization (€)**
Personal computer	2	2200	6	10.04
TOTAL	**-**	**-**	**-**	**10.04 €**

Table 3: *Software resources costs*

Software resources				
Product	**Units**	**Price/unit (€)**	**Useful life (years)**	**Amortization (€)**
macOS High Sierra	1	0	6	0
Google Chrome	1	0	-	0
LaTex	1	0	-	0
TOTAL	**-**	**-**	**-**	**0 €**

11.1.2 Data gathering

Table 4: *Human resources costs*

Human resources					
Role	**Units**	**Salary/hour (€)**	**Hours/week**	**Weeks**	**Cost (€)**
Project manager	1	35	20	2	1400
Software developer	1	35	20	2	1400
TOTAL	**-**	**-**	**-**	**-**	**2800 €**

Table 5: *Hardware resources costs*

Hardware resources				
Item	**Units**	**Price/unit (€)**	**Useful life (years)**	**Amortization (€)**
Personal computer	2	2200	6	20.08
TOTAL	**-**	**-**	**-**	**20.08 €**

Table 6: *Software resources costs*

Software resources				
Product	**Units**	**Price/unit (€)**	**Useful life (years)**	**Amortization (€)**
macOS High Sierra	1	0	6	0
Google Chrome	1	0	-	0
LaTex	1	0	-	0
Python	1	0	-	0
Public API's	N	0	-	0
TOTAL	**-**	**-**	**-**	**0 €**

11.1.3 Data preprocessing

Table 7: *Human resources costs*

Human resources					
Role	**Units**	**Salary/hour (€)**	**Hours/week**	**Weeks**	**Cost (€)**
Project manager	1	35	20	1	700
Software developer	1	35	20	1	700
TOTAL	**-**	**-**	**-**	**-**	**1400 €**

Table 8: *Hardware resources costs*

Hardware resources				
Item	**Units**	**Price/unit (€)**	**Useful life (years)**	**Amortization (€)**
Personal computer	2	2200	6	10.04
TOTAL	**-**	**-**	**-**	**10.04 €**

Table 9: *Software resources costs*

Software resources				
Product	**Units**	**Price/unit (€)**	**Useful life (years)**	**Amortization (€)**
macOS High Sierra	1	0	6	0
Google Chrome	1	0	-	0
LaTex	1	0	-	0
Python	1	0	-	0
TOTAL	**-**	**-**	**-**	**0 €**

11.1.4 Development of trading strategy

Table 10: *Human resources costs*

Human resources					
Role	**Units**	**Salary/hour (€)**	**Hours/week**	**Weeks**	**Cost (€)**
Project manager	1	35	15	7	3675
Data scientist	1	50	20	7	7000
TOTAL	**-**	**-**	**-**	**-**	**10675 €**

Table 11: *Hardware resources costs*

Hardware resources				
Item	**Units**	**Price/unit (€)**	**Useful life (years)**	**Amortization (€)**
Personal computer	2	2200	6	60.24
TOTAL	**-**	**-**	**-**	**60.24 €**

Table 12: *Software resources costs*

Software resources				
Product	**Units**	**Price/unit (€)**	**Useful life (years)**	**Amortization (€)**
macOS High Sierra	1	0	6	0
LaTex	1	0	-	0
Python	1	0	-	0
TOTAL	**-**	**-**	**-**	**0 €**

11.1.5 Total direct costs

Table 13: *Total direct costs*

Total direct costs	
Concept	Amount (€)
Market selection	
Human resources	1200
Hardware resources	10.04
Software resources	0
Subtotal	**1210.04**
Data gathering	
Human resources	2800
Hardware resources	20.08
Software resources	0
Subtotal	**2820.08**
Data preprocessing	
Human resources	1400
Hardware resources	10.04
Software resources	0
Subtotal	**1410.04**
Development of trading strategy	
Human resources	10675
Hardware resources	60.24
Software resources	0
Subtotal	**10735.24**
TOTAL	**16175.4 €**

11.2 Indirect costs

The two main indirect costs that this project has are the electricity and Internet. To obtain an approximation to the cost derived from the electricity, it is used the following data:

- Price of the Kwh in Spain: 0.12 €

- Total usage of the laptops: 405
- Power of the laptop: 61 W = 0.061 kW

It gives a cost of:

$$0.061kWh * 405h * 0.12e = 2.96euros$$

For the Internet, it is paid 45 €per month. We calculate the amortization based on that we usually use it 4 hours per day: (45*monthly_working_hours)/(30*4)

Table 14: *Total indirect costs*

Total indirect costs	
Type	Amount (€)
Electricity	2.96
Internet	60
TOTAL	**62.96 €**

11.3 Contingency costs

The contingency has been calculated as the 15% of the direct and indirect costs, since the planification has a reasonably high level of detail and we don't expect many deviations.

11.4 Incidental costs

Table 15: *Incidental costs*

Incidental costs				
Causes	Solution	Risk	Impact on cost (€)	Cost (€)
The data we decide to use is biased, and due to that the models cannot perform well	Buy unbiased data	25%	100	25
The selected API(s) for obtaining the data are not reliable or take too long to respond	Have redundancy in data gathering	15%	30	4.5
The models chosen to drive the trading strategy take a huge amount of time to get trained, to the point of making impossible an effective experimentation	Contract a cloud computing plan to train the models in the cloud	30%	40	12
TOTAL				**41.5 €**

11.5 Total cost

The total cost is not expected to increase beyond contingency because we have specified and quantified all the possible scenarios we can eventually face during the development of the project. Furthermore, depending on the level of expertise of the Developer and the Project Manager, the number of hours required to achieve the tasks can be less than the planned.

Table 16: *Total costs of the project*

Total costs	
Type	Amount (€)
Direct	16175.4
Indirect	62.96
Contingency	2435
Incidental	41.5
TOTAL without VAT	18714
TOTAL with 21% VAT	**22644.5 €**

11.6 Control Management

Control management needs to be done regularly during the development of the project, and it refers to some indicators that are able to evaluate how the project is fitting the schedule and budget. Some indicators that could be used in this project are the following:

- Task time consuming deviations: (estimated hours - actual hours)
- Cost deviations related to deviations in time (estimated hours - actual hours) * cost per hour

The proposed regularity for this indicators to be evaluated is at the end of every task (in order to evaluate the actual time consumed by it) and at the end of each month (just in case a task takes too long to conclude).

12 Sustainability report

12.1 Environmental

The environmental effect that will have the development of this project is estimated to be very low since all that is needed is a pair of laptops and a bit of electricity. One way in which the impact can be lowered even more is to use the laptops that we already have, instead of buying any new for this project. This option has been considered and will be implemented.

However, we're concerned that once the project will be deployed, depending on the chosen trading strategy, the system would require to update the data very frequently, and for example, re-train the models, which is an expensive operation, specially when we're using Deep Learning.

12.2 Economic

It has been presented a detailed planned and an estimated costs for the project, where it has been included human and material resources.

Since this is a project which aim is to optimize investments, it is estimated to have a positive economic impact on those who use it. Nevertheless, we have to bear in mind that if the product of this project is misused, or used irresponsibly (for example if someone trades with money that they actually need to pay the bills), the effects could happen to be eventually very negative.

12.3 Social

This project, as well as every project that proposes any kind of automatization, has its benefits and its drawbacks. In one hand, it could be beneficial for an institutional trader to get rid of some of its analysts if they can't perform better than the algorithms included in the product of this project. On the other hand, it is clear the possibility of destroying jobs. But why should a human do a task that can be done by a computer? I think humans should focus on those jobs that cannot be automated yet.

If we analyze how will it increase the life quality of the average retail trader, we find that the product of this project will free them from having to be all day in front of the screen watching the graphs move and the prices

going up and down. This is particularly true if the chosen strategy is based in finding the best possible moment to enter the market.

13 Market selection and data gathering

Before starting the development of the trading strategy, the market in which the trading will take place has to be chosen. To evaluate which would be the best fit for the project, the relevant aspects that have been taken into account are the following: **data availability** (both historical past data and real-time), **buy and sell volume**, i.e. if it is a market that has a relevant weight and has intense activity, and **time availability**, i.e. trading hours per week.
The markets that have been considered for this comparison are the **stock market**, the **Foreign Exchange**, and the **Exchange-Traded Funds** market.

13.1 Stock market

In the stock market it is relatively easy to obtain data about the current prices, either using an API or scrapping any finance website. Nonetheless, collecting past data is not as easy since many databases that one can find come with survivorship bias, and the ones that do not, are expensive and hard to find.

The trading volume is moderately high, but does not reach Forex levels. The time availability is reduced (unlike the values that have an uninterrupted market, stocks are traded on an specific schedule and on specific days).

For example, the Spanish Stock Exchange, IBEX35, has a schedule from Monday to Friday from 9:00 A.M. to 5:30 PM. This makes it not an optimal option for an algorithm that is operating 24/7 since during the closing times it would be stopped.

13.2 Forex Exchange

With regard to the currencies market, since it is the biggest market in trading volume and liquidity, the data availability is very convenient (both past data and real-time data). Therefore, it is a good candidate to be chosen

to do algorithmic trading. The schedule is continuous from Sunday night until Friday night, so we only have to have the algorithm stopped for a few hours on the weekend.

13.3 Exchange Traded Funds

The Exchange Traded Funds work very similarly to the stock market. Each fund's share has the value of the underlying assets, which are usually replicating some Index or diversified in a sector or region. The availability of real-time data is acceptable, but if we want to have historical data it is quite difficult to get it. The time availability is the same as that of the stock market, and the trading volume is significantly lower.

Once the previous alternatives have been analyzed, it has been decided to opt for the currency market (Forex), for having the best availability (time and data) and also the greater volume.

13.4 Data Gathering

In order to obtain the data required to train the models and backtest, some data sources have been considered. Note that there is a double requirement here: the past data and the real-time data.

For the past data, the chosen source has been a website called HistData[22], that offers minute data of almost any currency pair, and particularly from the one that will be used in this project: EUR/USD.

For the real-time data that will be used in the live trading, the elected data source has been an API service called Alpha Vantage[23], that offers real-time data on Forex and other markets, at no cost, and up to 5 queries per minute.

14 Knowledge integration

I have applied the required knowledge for solving problems using predictive models, acquired in the course **Aprenentatge Automàtic**, present in the Computer Science specialty, in the part of the project consisting of the

development of a Trading Strategy based on Machine Learning.

Also, I have applied the required knowledge for dealing with and gathering big amounts of data, acquired in the course **Cerca i Anàlisi d'Informació Massiva**, present in the Computer Science specialty, in the part of the project that consists of obtaining the financial data from the Internet and storing it in a reasonable way.

Thirdly, I have also integrated the knowledge acquired in the common course **Llenguatges de Programació**, in order to thoroughly evaluate which are the most reasonable programming languages to implement the project.

Finally, and this time I am referring to a non-technical competence, I have applied the acquired knowledge about economy, finance and markets obtained in the course **Empresa i Entorn Econòmic** in the formulation of the project and in the application of the technical knowledge.

15 Justification of project specialty

This project meets the requirements to be a Computation specialty TFG because it faces a problem that has to do with Computational Intelligence, Data Analysis, and Algorithmics. Furthermore, the problem is complex enough to require a deep understanding and an effective application of the previously described areas that are part of the Computation specialty.

Another reason to be considered is that all the good practices learned in Computation are planned to be applied in order to conduct a rigorous research and a meaningful experimentation.

To sum up, I think that this project has enough entity to be a Computation TFG because of the complexity of the problem to solve, the suggested techniques in order to do it, and all the background required in order to 1) Understand what is going on, and 2) Identify possible flaws and interpret the obtained results.

16 Technical competences and achievement level justification

CCO1.1: *Avaluar la complexitat computacional d'un problema, conèixer estratègies algorísmiques que puguin dur a la seva resolució, i recomanar, desenvolupar i implementar la que garanteixi el millor rendiment d'acord amb els requisits establerts.* **[Bastant]**

This competence is going to be achieved during the development of the Trading Strategy. The problem to solve is Trading, and the different algorithmic solutions to it will be deeply discussed. I have chosen the "Bastant" achievement level because I think it fits pretty well the scope of this project, where there is not a "perfect" solution, but a wide range of equally valid options with its pros/cons.

CCO2.1: *Demostrar coneixement dels fonaments, dels paradigmes i de les tècniques pròpies dels sistemes intel·ligents, i analitzar, dissenyar i construir sistemes, serveis i aplicacions informàtiques que utilitzin aquestes tècniques en qualsevol àmbit d'aplicació.* **[Una mica]**

I have chosen this competence because one of the goals of the project is to build an Automated Trading System that relies on an intelligent agent to operate, and this virtually what the competence describes. The chosen level is "Una mica" because the main focus of interest of this project is not to build a platform, but to experiment with the models instead.

CCO2.2: *Capacitat per a adquirir, obtenir, formalitzar i representar el coneixement humà d'una forma computable per a la resolució de problemes mitjançant un sistema informàtic en qualsevol àmbit d'aplicació, particularment en els que estan relacionats amb aspectes de computació, percepció i actuació en ambients o entorns intel·ligents.* **[Bastant]**

The reason behind the choice of this competence is that in order to feed the models and act on the market, the system will need to obtain, process, and consume data in a way that emulates a human-like behavior. So, programming a rule-based trading strategy it is a good representation and formalization of human knowledge. I have chosen the "Bastant" level because I think this is one of the core aspects of the project.

CCO2.4: *Demostrar coneixement i desenvolupar tècniques d'aprenentatge computacional; dissenyar i implementar aplicacions i sistemes que les utilitzin, incloent les que es dediquen a l'extracció automàtica d'informació i coneixement a partir de grans volums de dades.* **[En profunditat]**

I believe this is the competence that better describes the project, and this is why I have chosen it with the "En profunditat" level. Firstly, because the whole project is about the development of computational learning techniques, and secondly because it will require to access, process and consume a very big amount of data in order to perform well.

17 Identification of laws and regulations

17.1 Software licenses

The first regulation that directly affects the project are the licenses of the software that has been used during its development. The main piece of software that has been used is the Python programming language, that is distributed under a special license that they have called **GPL-compatible**. In general, it is very similar to GPL, with the difference that anyone can distribute a modified version of it without open sourcing the changes.
Given that modifying the language and distributing it are activities completely out of the scope of this project, it is considered that the use of this software is compliant with the applicable regulation.

17.2 Data usage

The next part of the project that can be affected by regulations is the data obtention and manipulation. In the case of the past data, it has been obtained via a website (histdata.com) that offers it without any kind of license nor regulation.
When it comes to the service chosen to consume the data in real time, it does have a User Terms and Conditions agreement, that has been carefully read. The most relevant part, or the one that could directly affect the project is the Intellectual Property. In short, this regulation explains that

any use of the data is permitted as long as it does not get redistributed. Given that the redistribution of the data is completely out of the scope of this project, it is considered that this regulation is not violated in any way.

17.3 Capital Markets and Investment regulations (Spain)

If we wanted to deploy the project in a way that it would be a portfolio management tool, it would have to be compliant with the legislation of the Spanish **Comisión Nacional del Mercado de Valores (CNMV)**. In particular, there is a subsection that refers to FinTech products (technological and financial), and establishes that, a company must be regulated, supervised and registered on the CNMV if:

1. Performs financial advisory activities and/or automated portfolio management
2. Performs Multi Account Management and/or Percentage Allocation Management
3. Sells or obtains any kind of profit of trading strategies created by third-party traders (Social trading)

Otherwise, if none of the above activities are performed and it is just the software/algorithms that is distributed, there is no requirement from the CNMV for the distributing company to be registered and supervised.

18 Similar or related products

18.1 AlgoTrader

AlgoTrader[13] is a Java-based desktop application (Windows-only) that allows users with little or no experience with programming to develop, backtest and execute trading strategies. It comes with a Graphical User Interface that makes it easier for unexperienced users because they can program the strategies using drag and drop actions and see the backtesting results in an Excel spreadsheet.

19 Development of the Trading Strategy

In this part of the project, the goal is to develop a trading strategy that makes use of Machine Learning models in order to decide whether is it worth to open a position. In order to measure the performance of this strategy, another simpler, commonly-used trading strategies will also be implemented to serve as benchmark. Finally, in order to measure how all these trading strategies perform when using real data, a backtesting platform will be implemented.

19.1 Exploration of the dataset

Before implementing any of the above, it may be interesting to get some insights about how the data looks like.

The dataset we are working on is about the price of the currency pair EUR/USD, that is, how many US Dollars one need to purchase 1 Euro. The dataset contains hourly information starting from the year 2000 to 2018, and it can be visualized on Figure 3.

This dataset has been assembled appending monthly csv files that contained price data with resolution of 1 minute. The webpage where they have been collected is HistData.com[22]

Figure 3: *Line plot of the EUR/USD prices dataset*

19.2 Development of rule-based trading strategies

19.2.1 Random Trading Strategy

This trading strategy has been done for comparison purposes. For simplicity it is allowed only an open position at a time.
The entry condition is that there are not open positions, and the side is decided randomly. The positions are closed according to 3 input parameters: a Stop Loss, a Take Profit and a Time Limit.

19.2.2 Mean Reversion Trading Strategy

This strategy is based on the rule that states that the price of an asset tends to evolve towards the mean of the past N observations, with N being large enough.
The input parameters are: the window size (how many past observations are taken into account for computing the mean), and as in the previous strategy, a Stop Loss, a Take Profit and a Time Limit to close the positions.

19.2.3 Trend Following Trading Strategy

The third strategy that has been implemented is based on identifying trends and following them.
The trend is identified in the following way: If the absolute value of the mean of the first-differences of the past N observations is greater that a certain threshold, then a trend is identified. The input parameters for this strategy are: the window size, the threshold, the Stop Loss, Take Profit and Time Limit.

19.3 ML approach: Long-Short Term Memory Network

Finally, after implementing the traditional strategies, the trading strategy that is powered by a Machine Learning model has been implemented. The procedure in which operates is the following: A new column has been added to the dataset. It contains the prediction that the network has made for the next time horizon. In each price update, the current price is compared to the prediction and, if there are not open positions, a sell position is placed if the prediction is lower, and a buy position is placed otherwise.

19.3.1 Brief review of Recurrent Neural Networks

To better understand how LSTMs work, first it is convenient to review the behavior of the Recurrent Neural Network.

First, the input data get transformed into machine-readable vectors. Then, the RNN processes the sequence of vectors one by one. While processing, it passes the previous hidden state to the next step of the sequence. The hidden state acts as the neural networks memory. It holds information on previous data the network has seen before.

Consider a cell of the RNN to see how the hidden state would be computed. First, the input and previous hidden state are concatenated to form a vector. That vector now has information on the current input and previous inputs. The vector passes through the tanh activation, that squishes the values between -1 and 1, and the output is the new hidden state, or the memory of the network.

Due to its simplicity, RNNs use a lot less computational resources than it's evolved variants, LSTMs and GRUs.

19.3.2 What is an LSTM Network?

Long short-term memory (LSTM) units (Figure 4) are units of a recurrent neural network (RNN). An RNN composed of LSTM units is often called an LSTM network (or just LSTM). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

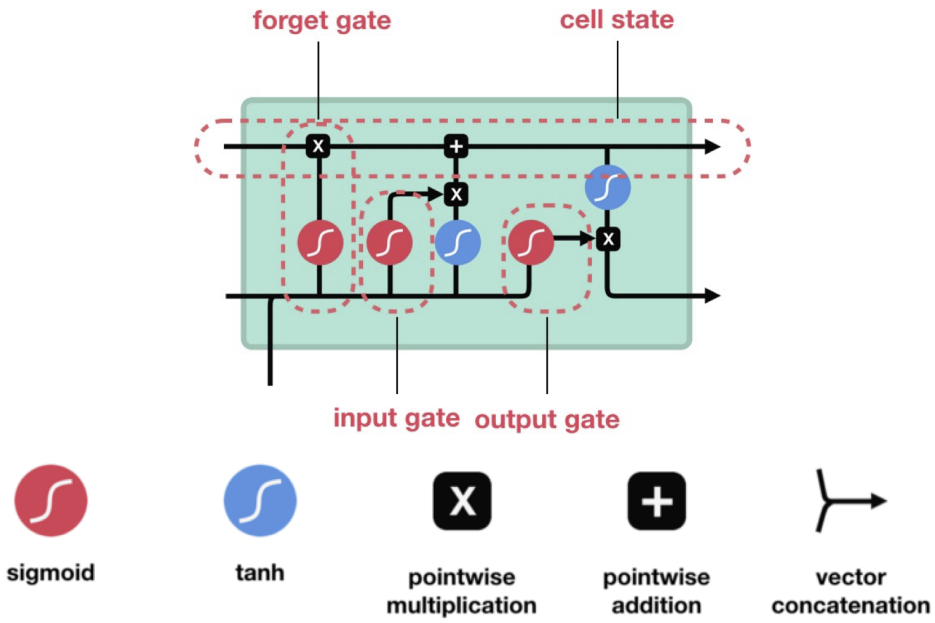

Figure 4: *Diagram of an LSTM Unit[15]*

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.[24]
The graphics and information about LSTMs that have been used in this section have been based on the work by M. Nguyen (2018)[15].

19.3.3 Core concept

The core concept of LSTMs is the cell state, and its various gates. The cell state act as a transport highway that transfers relative information all the way down the sequence chain. It can be seen as the "memory" of the network. The cell state, in theory, can carry relevant information throughout the processing of the sequence. So even information from the earlier time steps can make its way to later time steps, reducing the effects

of short-term memory. As the cell state goes on its journey, information gets added or removed to the cell state via gates. The gates are different neural networks that decide which information is allowed on the cell state. The gates can learn what information is relevant to keep or forget during training. A Python pseudo code of the LSTM cell can be found in Figure 5.

19.3.4 Input Gate

To update the cell state, there is an input gate. First, the previous hidden state and current input are passed through a sigmoid function. That decides which values will be updated by transforming the values to be between 0 and 1. It's also passed the hidden state and current input into the tanh function to squish values between -1 and 1 to help regulate the network. Then, the tanh output is multiplied by the sigmoid output. The sigmoid output will decide which information is important to keep from the tanh output.

19.3.5 Cell State

Combining the forget and input gates output, there is enough information to calculate the cell state. First, the cell state gets pointwise multiplied by the forget vector. This has a possibility of dropping values in the cell state if it gets multiplied by values near 0. Then, the output from the input gate contributes with a pointwise addition which updates the cell state to new values that the neural network finds relevant.

19.3.6 Output Gate

The output gate decides what the next hidden state should be. Note that the hidden state contains information on previous inputs. The hidden state is also used for predictions. First, both the previous hidden state and the current input are passed through a sigmoid function. Then we pass the newly modified cell state to the tanh function. We multiply the tanh output with the sigmoid output to decide what information the hidden state should carry. The output is the hidden state. The new cell state and the new hidden is then carried over to the next time step.

```
def LSTMCELL(prev_ct, prev_ht, input):
    combine = prev_ht + input
    ft = forget_layer(combine)
    candidate = candidate_layer(combine)
    it = input_layer(combine)
    Ct = prev_ct * ft + candidate * it
    ot = output_layer(combine)
    ht = ot * tanh(Ct)
    return ht, Ct



ct = [0, 0, 0]
ht = [0, 0, 0]

for input in inputs:
    ct, ht = LSTMCELL(ct, ht, input)
```

Figure 5: *LSTM cell pseudo code[15]*

19.3.7 Splitting the dataset: Training and Test sets

Since we want the predictions to be the least overfitted possible, a split of the dataset has been made.
The training data conforms an eighty percent of the total amount of data and the test data the other twenty percent. Furthermore, the test data has been chosen to be the last part of the sequences of prices, since we are more interested in checking the validity of the model in prices that are closer to the present moment.
To sum up, the training set is formed by the sequence of prices that goes from the year 2000 to the year 2016, and the test set from there to the last observed price (November 2018). The train/test split can be visualized in Figure 6.

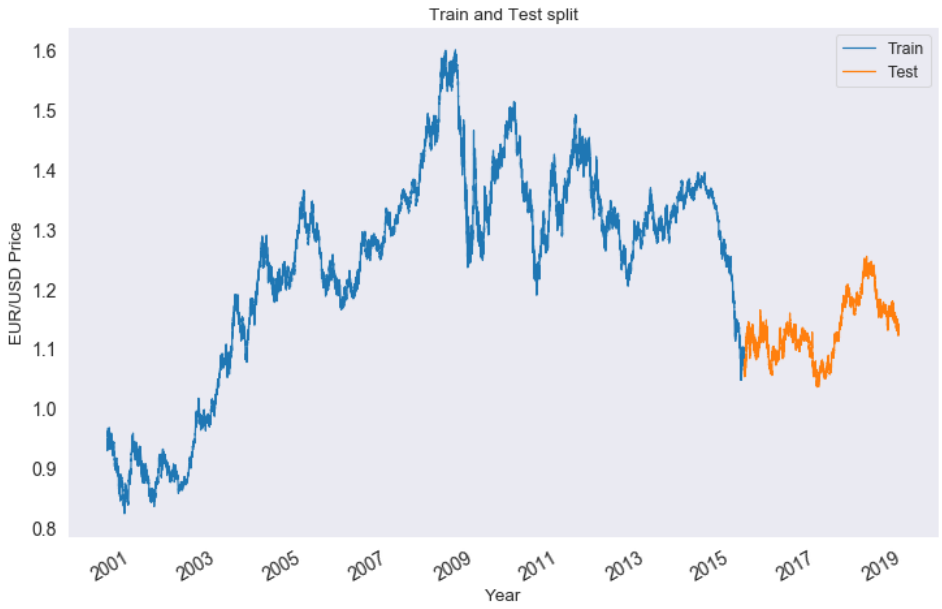

Figure 6: *Splitted dataset into train and test sets*

19.3.8 Deciding the hyper-parameters of the LSTM

In order to optimize the performance of the model, a grid search of the hyperparameters has been made. The hyperparameters that have been taken into account are:

- Number of units (or LSTM cells) – Range: [96, 120, 150]
- Size of the input sequence (i.e. sliding window) – Range: [25, 50, 100]
- Number of epochs in training – Range: [25, 50]
- Batch size – Range: [32, 64, 128]

All the different combinations of these hyperparameters have been tested out, with the intention of selecting the combination that leads to the better performance. For each hyperparameter tuple, an LSTM network has been trained using these hyperparameters and the training set. Then, to discriminate the better tuple, every network has computed the outputs of the test set, and the Rooted Mean Squared Error metric has been applied with respect to the ground truth, with the intention of keeping the tuple that yields the lesser cost in the test set.
The top 25 results of this search can be seen in Table 17.

Table 17: *Hyperparameter grid search top 25 results*

Batch Size	Sequence lenght	Epochs	Units	RMSE
32	25	50	150	0.02052
32	25	50	96	0.02097
32	50	50	150	0.02219
32	100	50	120	0.02259
32	50	50	120	0.02289
64	25	50	150	0.02308
32	25	50	120	0.02318
32	100	50	96	0.02327
32	50	50	96	0.02334
64	25	50	120	0.02438
32	100	50	150	0.02706
32	50	20	96	0.02727
64	50	50	150	0.0273
32	50	20	150	0.02833
128	25	50	150	0.02873
32	100	20	150	0.02992
32	25	20	96	0.03036
64	100	50	120	0.03072
32	25	20	150	0.03091
128	50	50	96	0.03097
128	100	50	96	0.03193
128	100	50	120	0.03207
64	25	20	120	0.03262
64	100	50	150	0.03364
128	50	50	150	0.03397

19.3.9 Training the LSTM Network

As shown in the previous section, an experiment to select the hyperparameters that lead to better results has been carried out. The final model has been then trained having this information in mind. A plot showing the predictions versus the actual price can be seen in the Figure 7.

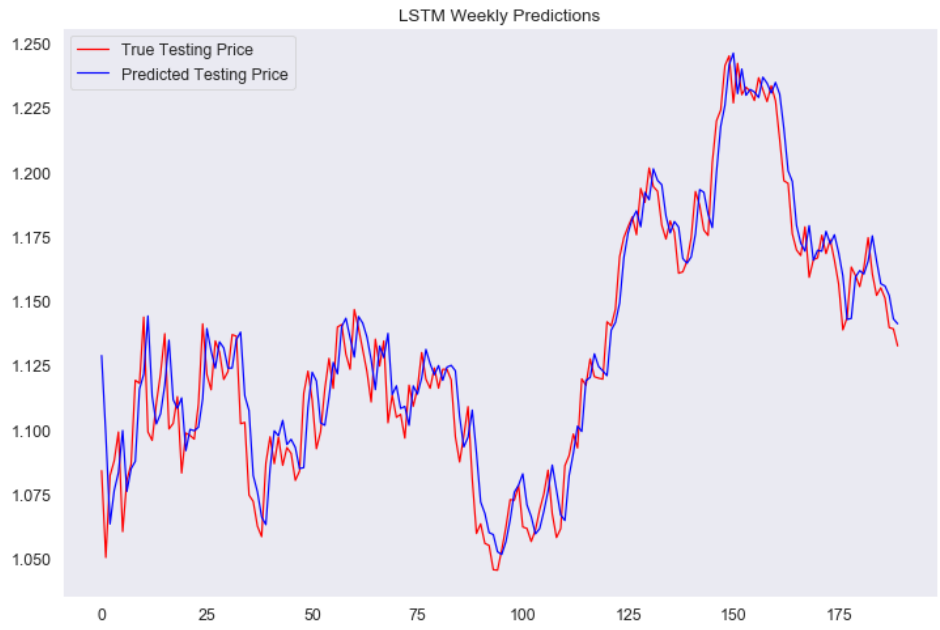

Figure 7: *Predictions made with LSTM on weekly data*

19.3.10 Evaluating the model

Seeing the plot in the Figure 7, one can think the model has performed pretty well, but in order to get some deeper insights about the evaluation of the model, the rooted mean squared error metric has been computed and also a plot of the distribution of the residuals (Figure 8).
The Rooted Mean Squared Error metric for this model evaluated versus the test set is **0.02052**.
In the plot of the distribution of the residuals also can be seen that the residuals are centered at 0 and the errors follow a Gaussian-like distribution.

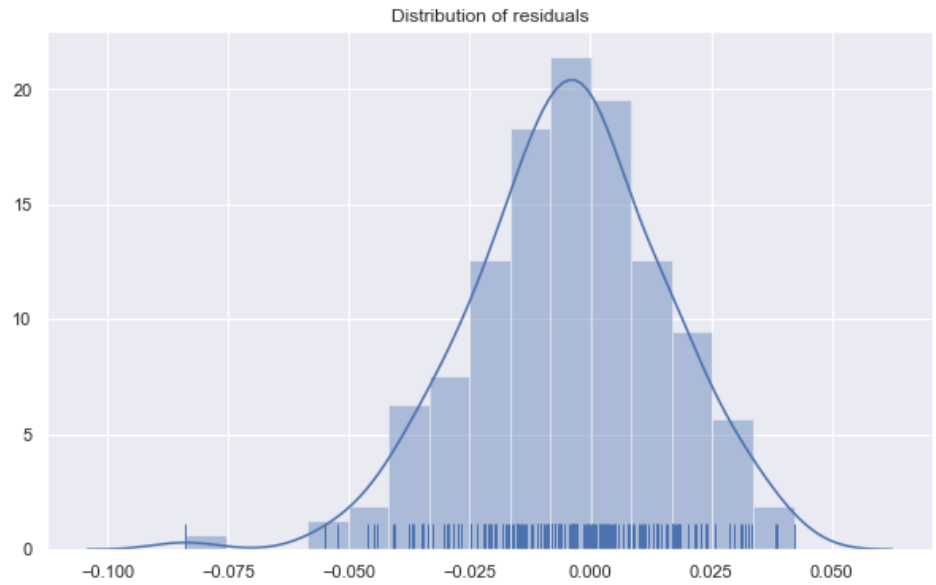

Figure 8: *Distribution of the residuals in the LSTM price prediction*

19.3.11 Implementing the LSTM Trading Strategy

After having predicted the price for every single week on the test set, the next challenge is to integrate these predictions in an actual trading strategy.

The first approach that was thought of was to make the prediction for the next week in each new price that arrives. But given that in this project the dataset is static, this would be highly inefficient.

Instead, the solution that has been finally implemented is the following: The predictions made in the test set for every week, have been added as a column of the dataframe, so with each new price that enters the backtesting system, it comes with the next week forecast. This way, it is easier to compare the current price with the prediction and operate accordingly.

Particularly, the strategy implemented here is very simple: If the current price is lower than the forecasted for the following week, place a buy position. And the same in the opposite case: place a sell position if the forecast for the next week is lower than the current price.

20 Results

20.1 Rule-based trading strategies

For each rule-based trading strategy, a grid search of the parameters has been done in order to select which are the ones that lead to a more profitable strategy.

20.1.1 Random trading strategy

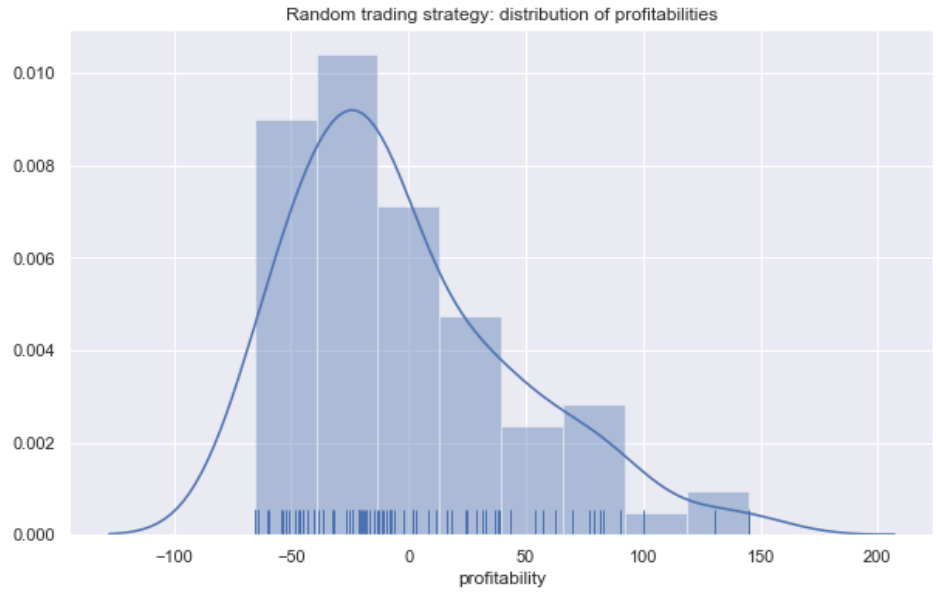

Figure 9: *Random trading strategy: distribution of profitabilities*

As it can be seen in the histogram (Figure 9), the kernel density estimation of the profitabilities of this strategy when using different combinations of stop loss, take profit and time limit is centered at the left of 0. This means that the majority of the experiments that have been conducted using this strategy have resulted with a lower amount of money than the starting amount (i.e. they have been losing strategies in the long term).

Nonetheless, and due to the apparently random nature of the market, there have been some winning experiments, even one that has achieved a profitability of 150%.

20.1.2 Mean reversion strategy

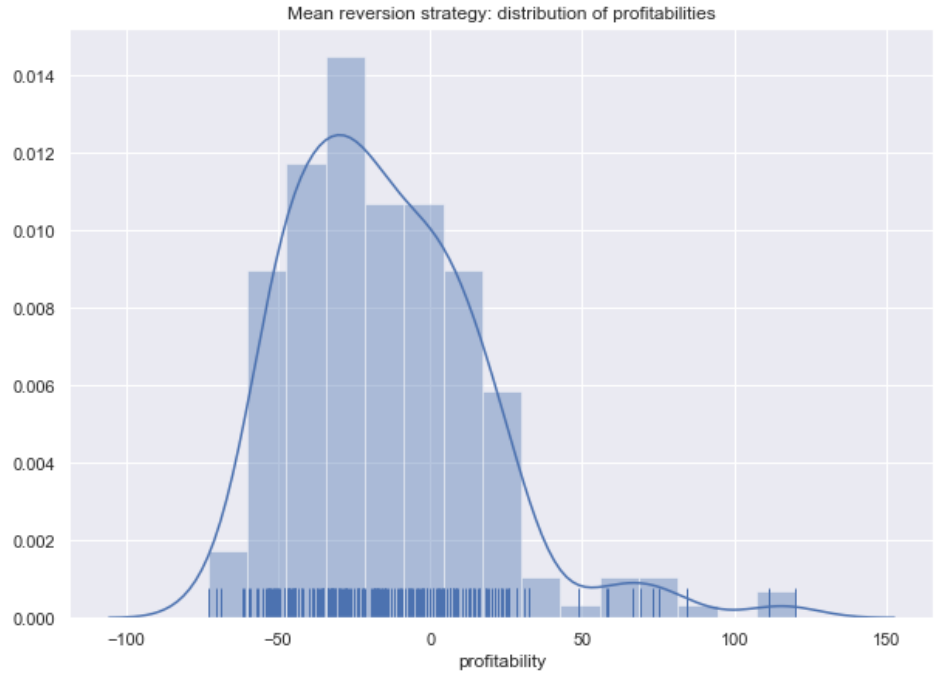

Figure 10: *Mean reversion strategy: distribution of profitabilities*

For the mean reversion strategy, the obtained results have been pretty similar to the random strategy. The profitabilities histogram (Figure 10) shows that the kernel density estimation is centered in the part where the profitability is negative, but there are some winning experiments.

This results have been quite unexpected to me because I thought that this strategy would perform way better than the random one, since this one uses historical data and it does a more informed operation.

One limitation that could have affected the performance of this strategy is the time scale: remember that this strategy is based in the belief that everything that goes up must come down, but it remains unknown whether if it comes down in a day, in a week, or in a decade. Thus, since in this experiments the operations have not been particularly focused on the long term investment, this rule can be unmanifested in the time scale of the operation of this strategy.

20.1.3 Trend following strategy

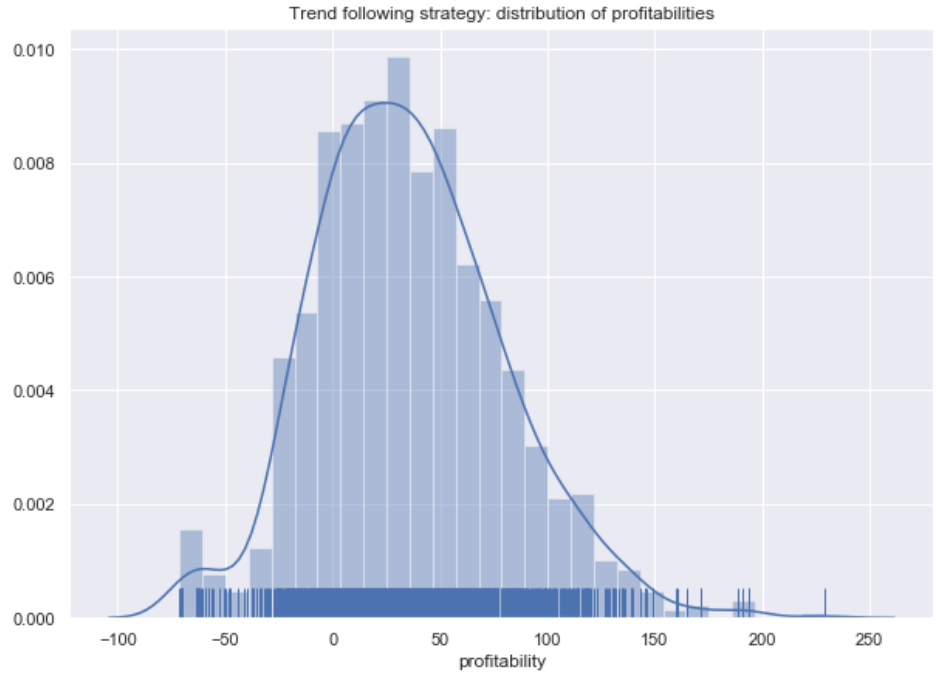

Figure 11: *Trend following strategy: distribution of profitabilities*

As it can be seen in the histogram (Figure 11), the profitabilities obtained using this strategy are higher than in the two previous strategies.

The kernel density estimation is centered at the positive profitabilities part and the most performing experiment have resulted in a profitability of 230%, originated from this parameter configuration: a slope threshold of 2.27e-05, a stop loss of 0.05, a take profit of 0.15, a time limit of 1344 hours, and a window size of 496 hours.

One limitation that can be found in the application of this particular strategy is that, although the beginning of a trend is supposedly well identified, due to the simplicity of the backtesting system used in this project, there is no way of identifying the end of a trend, being the 3 only close conditions the stop loss, the take profit and the time limit.

20.1.4 LSTM trading strategy

Since the hyperparameter optimization for the LSTM that this strategy uses has already been done in a previous section, and given that the predictions are only one step ahead, it makes no sense to try different configurations of the parameters that close the operations, because one operation will be open at each price update, and closed in the next one.

The profitability obtained using this strategy has been **-79.25%**. That is, worse than the worst parameter configuration in the previous 3 strategies. Even worse than the most unlucky random trading strategy.

20.2 Interpretation of LSTM trading strategy results

The purpose of this section is to try to explain the obtained results, and see which are its possible causes.

20.2.1 A closer look to the plots

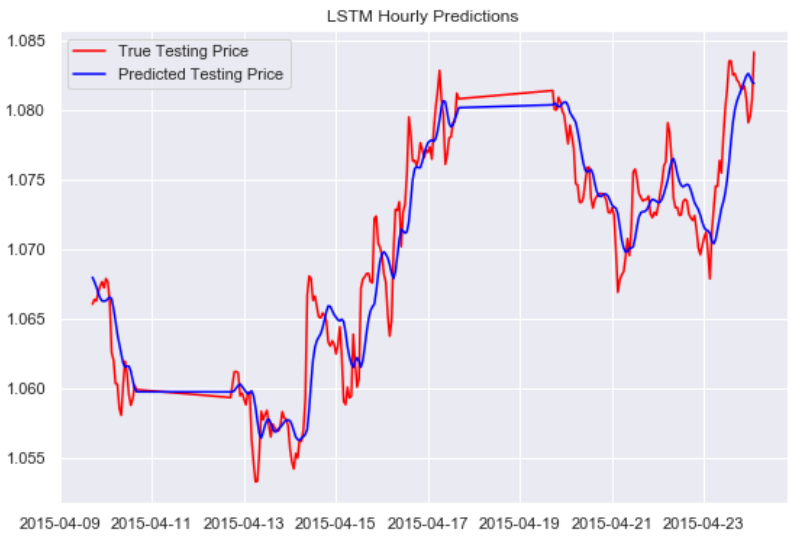

Figure 12: *First 250 prices and predictions of the test set*

In the Figure 12, can be seen that movements that the predictions make are slightly delayed from respect the actual movements of the market. Thus, the forecast that the LSTM provides in this case is not useful because

it arrives late. This behavior can also be seen on figure 7, but with weekly predictions instead.

20.2.2 Ups and downs

In essence, the key information that the LSTM strategy uses to guide its operation is whether the price will go up or down in the next period. The exact value is not important for the decision. So, in order to test if the predictions at least are correct in this sense, the percentage of good predictions has been calculated.

The results have been the following: 51% of the time, the direction of the next price has been correctly identified, a total of 11493 times, versus 10966 times in which it was incorrectly identified. This is not any better than the accuracy that one would get in predicting a bunch of coin flips.

20.2.3 Correlation test

This test will examine how correlated are the predictions on the percentage change with respect to the actual observed percentage change in the price. The obtained correlation coefficient is **0.08**, and in the plot (Figure 13), any proper correlation is seen. The conclusion extracted from this test is that, since the correlation coefficient is so small, the predictions made by the LSTM do not reflect the changes in the price, and thus they do not constitute a reliable indicator of the future evolution of the price.

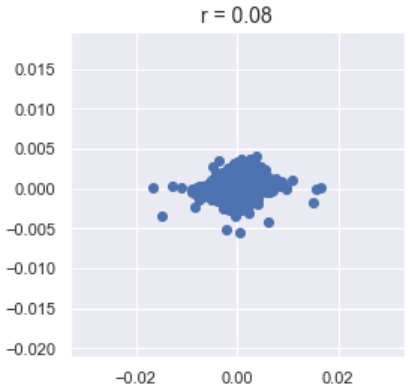

Figure 13: *Correlation in the percentage change*

21 Conclusions

1. The first conclusion of the work conducted here is that the price of an asset cannot be reliably forecasted by an LSTM Network using only its historical price. Indeed, the network is effectively able to learn. But it ends up using a strategy in which predicting a value close to the previous one turns out to be the most effective way of minimizing the loss function.

2. This project was conceived with the purpose of comparing the traditional trading strategies with the one derived from the predictions of a Machine Learning model. In some sense, before doing the experiments, I was biased towards the belief that the Machine Learning approach would beat the rule-based one, because I thought the model could learn all the simple patterns that traditional strategies use and combine them in order to maximize the performance. But surprisingly (to me) it happened exactly the opposite.

3. In practice, the results of one step ahead prediction models based on historic price data alone, as the one showcased here, remain hard to accomplish and are not particularly useful for trading.

4. Needless to say that more sophisticated approaches of implementing useful LSTMs for price predictions potentially do exist. Using more data, such as sentiment analysis or fundamental data, as well as optimizing network architecture are a starting point. In my opinion, however, there is more potential in incorporating data and features that go beyond historic prices alone. After all, the finance world has already known for long that “past performance is not an indicator for future outcomes”.

References

[1] Chan, Ernest P., "Quantitative Trading: How to build your own Algorithmic Trading business." Wiley, 2009.

[2] Chan, Ernest P., "Machine Trading: Deploying computer algorithms to conquer the markets." Wiley, 2017.

[3] Shobhit, S., "Basics of algorithmic trading: Concepts and examples" Online. URL: https://www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp. Visited on 20-09-2018

[4] Investopedia, "What is the Financial Market" Online. URL: https://www.investopedia.com/terms/f/financial-market.asp. Visited on 25-09-2018

[5] Investopedia, "What is the 'Commodity Market'" Online. URL: https://www.investopedia.com/terms/c/commodity-market.asp. Visited on 25-09-2018

[6] Wikipedia, "Automated Trading Systems" Online. URL: https://en.wikipedia.org/wiki/Automated_trading_system. Visited on 25-09-2018

[7] Morton Glantz, Robert Kissell. "Multi-Asset Risk Modeling: Techniques for a Global Economy in an Electronic and Algorithmic Trading Era". Academic Press, Dec 3, 2013, p. 258

[8] Teamgantt, online gantt chart software. https://www.teamgantt.com/. Online. Accessed: 29-09-2018.

[9] Salary.com, salary calculator per role. https://www1.salary.com/. Online. Accessed: 06-10-2018.

[10] The Python programming language. python.org

[11] The R Project for Statistical Computing. https://www.r-project.org/

[12] The C++ Programming Language. www.cplusplus.com/

[13] AlgoTrader: Algorithmic Trading Software. https://www.algotrader.com/

[14] Gated Recurrent Unit. https://arxiv.org/abs/1701.03452

[15] Long Short Term Memory. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[16] CNMV Regulation for FinTech. http://cnmv.es/QAsFinTech.pdf Visited on 25-11-2018

[17] Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer, ISBN 978-0-387-31073-2

[18] Investopedia: What is a Trading Strategy? https://www.investopedia.com/trading-strategy-4427764. Visited on 29-12-2018

[19] Investopedia: Automated Trading Systems: The Pros and Cons https://www.investopedia.com/articles/trading/11/automated-trading-systems.asp Visited on 29-12-2018

[20] Investopedia: Backtesting investopedia.com/backtesting.asp Visited on 29-12-2018

[21] Survivorship bias: http://www.scientificamerican.com/article/how-the-survivor-bias-distorts-reality/ Visited on 29-12-2018

[22] HistData.com: Free Forex Historical Data http://www.histdata.com/download-free-forex-data/ Visited on 23-11-2018

[23] Apha Vantage https://www.alphavantage.co/ Visited on 23-11-2018

[24] Long-Short Term Memory https://en.wikipedia.org/wiki/Long_short-term_memory Visited on 11-01-2019

[25] Article on LSTM asset price prediction https://hackernoon.com/dont-be-fooled-deceptive-cryptocurrency-price-predictions-using-deep-learning-bf27e4837151?gi=ed17c7e55d97. Visited on 29-11-2018

[26] Stock Prices Don't Predict Stock Prices https://medium.com/apteo/stock-prices-dont-predict-stock-prices-bbf3e421bedf. Visited on 29-11-2018